

# As Implemented EOS Metadata Model Definition

*Derived from the B.0 Descriptor Template (9/30/1998)*

## Collection Level Metadata

### Full Collection Level Metadata Coverage

```
1{ G = CollectionMetadata }1
 1{ O = Type, t = "string<20>" }1 +
 1{ O = DLLNamefile, t = "string<255>" }1 +
 1{ O = SpatialSearchType, t = "string<40>" }1 +
 1{ G = CollectionDescriptionClass }1 +
 1{ G = ECSCollection }1 +
 1{ G = Contact }1 +
 1{ G = DisciplineTopicParameters }1 +
 0{ G = SingleTypeCollection }1 +
 0{ G = Spatial }1 +
 0{ G = Temporal }1 +
 0{ G = TemporalKeywordClass }1 +
 0{ G = SpatialKeywordClass }1 +
 0{ G = Locality }1 +
 0{ G = ProcessingLevel }1 +
 0{ G = Platform }1 +
 0{ G = AnalysisSource }1 +
 0{ G = Campaign }1 +
 0{ G = CollectionAssociation }1 +
 0{ G = Review }1 +
 0{ G = CSDTDescription }1 +
 0{ G = AdditionalAttributes }1 +
 0{ G = StorageMediumClass }1
```

### Intermediate Collection Level Metadata Coverage

```
1{ G = CollectionMetadata }1
 1{ O = Type, t = "string<20>" }1 +
 1{ O = DLLNamefile, t = "string<255>" }1 +
 1{ O = SpatialSearchType, t = "string<40>" }1 +
 1{ G = CollectionDescriptionClass }1 +
 1{ G = ECSCollection }1 +
 1{ G = Contact }1 +
 1{ G = DisciplineTopicParameters }1 +
 0{ G = SingleTypeCollection }1 +
 0{ G = Spatial }1 +
 0{ G = Temporal }1 +
 0{ G = TemporalKeywordClass }1 +
 0{ G = SpatialKeywordClass }1 +
 0{ G = Locality }1 +
 0{ G = ProcessingLevel }1 +
 0{ G = Platform }1 +
 0{ G = AnalysisSource }1 +
 0{ G = Campaign }1 +
```

```

0{ G = CollectionAssociation }1 +
0{ G = Review }1 +
0{ G = CSDTDescription }1 +
0{ G = AdditionalAttributes }1 +
0{ G = StorageMediumClass }1

```

### **Limited or Minimal Collection Level Metadata Coverage**

```

1{ G = CollectionMetadata }1
 1{ O = Type, t = "string<20>" }1 +
 1{ O = DLLNamefile, t = "string<255>" }1 +
 1{ O = SpatialSearchType, t = "string<40>" }1 +
 1{ G = CollectionDescriptionClass }1 +
 1{ G = ECSCollection }1 +
 1{ G = Contact }1 +
 1{ G = DisciplineTopicParameters }1 +
 0{ G = SingleTypeCollection }1 +
 0{ G = Spatial }1 +
 0{ G = Temporal }1 +
 0{ G = TemporalKeywordClass }1 +
 0{ G = SpatialKeywordClass }1 +
 0{ G = Locality }1 +
 0{ G = ProcessingLevel }1 +
 0{ G = Platform }1 +
 0{ G = AnalysisSource }1 +
 0{ G = Campaign }1 +
 0{ G = CollectionAssociation }1 +
 0{ G = Review }1 +
 0{ G = CSDTDescription }1 +
 0{ G = AdditionalAttributes }1 +
 0{ G = StorageMediumClass }1

```

### **Collection Level Metadata Compound Definitions (Nest Level 2)**

```

1{ G = CollectionDescriptionClass }1
 1{ O = ShortName, t = "string<8>" }1 +
 1{ O = LongName, t = "string<80>" }1 +
 1{ O = CollectionDescription, t = "string<255>" }1 +
 1{ O = VersionID, t = "integer<1..255>" }1

```

```

1{ G = ECSCollection }1
 1{ O = RevisionDate, t = "date" }1 +
 0{ O = SuggestedUsage, t = "string<500>" }1 +
 0{ O = ProcessingCenter, t = "string<20>" }1 +
 1{ O = ArchiveCenter, t = "string<20>" }1 +
 1{ O = VersionDescription, t = "string<255>" }1

```

```

1{ G = Contact }1
 0{ G = ContactPerson }1 +
 1{ G = ContactOrganization }1

```

```

1{ G = DisciplineTopicParameters }1
 1{ C = DisciplineTopicParametersContainer }n

```

```

0{ G = SingleTypeCollection }1
 0{ O = CitationforExternalPublication, t = "string<255>" }1 +

```

```

1{ O = CollectionState, t = "string<255>" }1 +
1{ O = MaintenanceandUpdateFrequency, t = "string<80>" }1 +
0{ O = AccessConstraints, t = "string<255>" }1 +

0{ G = Spatial }1
 1{ O = SpatialCoverageType, t = "string<25>" }1 +
 1{ G = SpatialDomainContainer }1 +
 0{ G = CoordinateSystemContainer }1 +

0{ G = Temporal }1
 1{ O = TimeType, t = "string<10>" }1 +
 1{ O = DateType, t = "string<10>" }1 +
 1{ O = TemporalRangeType, t = "string<30>" }1 +
 1{ O = PrecisionofSeconds, t = "integer<>" }1 +
 1{ O = EndsatPresentFlag, t = "string<1>" }1 +
 1[ 1{ G = RegularPeriodic }1 |1{ G = MultipleDateTimePeriod }1 |1{ G = SingleDateTime }1 |1{ G =
RangeDateTime }1 ] 1

0{ G = TemporalKeywordClass }1
 1{ O = TemporalKeyword, t = "string<10>" }n

0{ G = SpatialKeywordClass }1
 1{ O = SpatialKeyword, t = "string<10>" }n

0{ G = Locality }1
 1{ C = LocalityContainer }n

0{ G = ProcessingLevel }1
 1{ O = ProcessingLevelDescription, type = string<80>" }1 +
 1{ O = ProcessingLevelID, t = "string<6>" }1

0{ G = Platform }1
 1{ C = PlatformContainer }n

0{ G = AnalysisSource }1
 1{ C = AnalysisSourceContainer }n

0{ G = Campaign }1
 1{ C = CampaignContainer }n

0{ G = CollectionAssociation }1
 1{ C = CollectionAssociationContainer }n

0{ G = Review }1
 1{ C = ReviewContainer }n

0{ G = CSDTDescription }1
 1{ O = PrimaryCSDT, t = "string<30>" }1 +
 0{ O = IndirectReference, t = "string<100>" }1 +
 1{ O = Implementation, t = "string<100>" }1 +
 0{ O = CSDTComments, t = "string<255>" }1

0{ G = AdditionalAttributes }1
 1{ C = AdditionalAttributesContainer }n

0{ G = StorageMediumClass }1
 1{ O = StorageMedium, string<30>" }1

```

### ***Collection Level Metadata Compound Definitions (Nest Level 3)***

```
0{ G = ContactPerson }1
```

```

1{ C = ContactPersonContainer }n
1{ G = ContactOrganization }1
 1{ C = ContactOrganizationContainer }n

1{ C = DisciplineTopicParametersContainer }n
 1{ O = ECSDisciplineKeyword, t = "string<24>" }1 +
 1{ O = ECSTopicKeyword, t = "string<32>" }1 +
 1{ O = ECSTermKeyword, t = "string<50>" }1 +
 1{ O = ECSVariableKeyword, t = "string<80>" }1 +
 0{ G = ECSParameter }1

1{ G = SpatialDomainContainer }1
 0{ G = VerticalSpatialDomain }1 +
 1{ G = HorizontalSpatialDomainContainer }1

0{ G = CoordinateSystemContainer }1
 0{ C0 = VerticalCoordinateSystemContainer }1
 1{ G = HorizontalCoordinateSystemContainer }1

1{ G = RegularPeriodic }1
 1{ C = RegularPeriodicContainer }n

1{ G = MultipleDateTimePeriod }1
 1{ C = MultipleDateTimePeriodContainer }n +

1{ G = SingleDateTime }1
 1{ O = TimeOfDay, t = "time" }1 +
 1{ O = CalendarDate, t = "date" }1 +

1{ G = RangeDateTime }1
 1{ O = RangeBeginningDate, t = "date" }1 +
 1{ O = RangeBeginningTime, t = "time" }1 +
 1{ O = RangeEndingDate, t = "date" }1 +
 1{ O = RangeEndingTime, t = "time" }1

1{ C = LocalityContainer }n
 0{ O = LocalityDescription, t = "string<255>" }1 +
 1{ O = LocalityType, t = "string<20>" }1

1{ C = PlatformContainer }n
 1{ O = PlatformShortName, t = "string<20>" }1 +
 1{ O = PlatformLongName, t = "string<80>" }1 +
 1{ O = PlatformType, t = "string<20>" }1 +
 0{ G = PlatformCharacteristic }1 +
 0{ G = Instrument }1

1{ C = AnalysisSourceContainer }n
 1{ O = AnalysisShortName, t = "string<20>" }1 +
 0{ O = AnalysisLongName, t = "string<80>" }1 +
 0{ O = AnalysisTechnique, t = "string<80>" }1 +
 1{ O = AnalysisType, t = "string<20>" }1

1{ C = CampaignContainer }n
 1{ O = CampaignShortName, t = "string<20>" }1 +
 1{ O = CampaignLongName, t = "string<80>" }1 +
 1{ O = CampaignStartDate, t = "date" }1 +
 1{ O = CampaignEndDate, t = "date" }1

1{ C = CollectionAssociationContainer }n

```

```

1{ O = CollectionType, t = "string<20>" }1 +
1{ O = CollectionUse, t = "string<500>" }1 +
1{ O = ShortName, t = "string<8>" }1 +
1{ O = VersionID, t = "integer<0..255>" }1

1{ C = ReviewContainer }n
 1{ O = ScienceReviewDate, t = "date" }1 +
 1{ O = ScienceReviewStatus, t = "string<20>" }1 +
 0{ O = FutureReviewDate, t = "date" }1

1{ C = AdditionalAttributesContainer }n
 1{ O = AdditionalAttributeDatatype, t = "string<10>" }1 +
 1{ O = AdditionalAttributeDescription, t = "string<255>" }1 +
 1{ O = AdditionalAttributeName, t = "string<40>" }1 +
 0{ G = PhysicalParameterDetails }1
 0{ G = InformationContent }1

```

#### ***Collection Level Metadata Compound Definitions (Nest Level 4)***

```

1{ C = ContactPersonContainer }n
 1{ O = Role, t = "string<16>" }1 +
 0{ O = HoursofService, t = "string<255>" }1 +
 0{ O = ContactInstructions, t = "string<255>" }1 +
 0{ O = ContactJobPosition, t = "string<255>" }1 +
 1{ O = ContactFirstName, t = "string<255>" }1 +
 0{ O = ContactMiddleName, t = "string<255>" }1 +
 1{ O = ContactLastName, t = "string<255>" }1 +
 0{ G = ContactPersonAddress }1
 0{ G = Telephone }1 +
 0{ G = Email }1

1{ C = ContactOrganizationContainer }n
 1{ O = Role, t = "string<16>" }1 +
 0{ O = HoursofService, t = "string<255>" }1 +
 0{ O = ContactInstructions, t = "string<255>" }1 +
 1{ O = ContactOrganizationName, t = "string<255>" }1 +
 0{ G = ContactOrganizationAddress }1 +
 0{ G = OrganizationTelephone }1 +
 0{ G = OrganizationEmail }1

0{ G = ECSParameter }1
 1{ O = ECSParameterKeyword, t = "string<24>" }n

0{ G = VerticalSpatialDomain }1
 1{ C = VerticalSpatialDomainContainer }n +

1{ G = HorizontalSpatialDomainContainer }1
 0{ G = ZoneIdentifierClass }1 +
 1[ 1{ G = GPolygon }1 | 1{ G = BoundingRectangle }1 | 1{ G = Point }1 | 1{ G = Circle }1 ]1

0{ C0 = VerticalCoordinateSystemContainer }1
 1{ G = AltitudeSystemDefinition }1 +
 1{ G = DepthSystemDefinition }1 +

1{ G = HorizontalCoordinateSystemContainer }1
 0{ G = GeodeticModel }1 +
 1[ 1{ G = GeographicCoordinateSystem }1 | 1{ G = PlanarCoordinateSystems }1 | 1{ G =
LocalCoordinateSystem }1 ]1 +

```

```

1{ C = RegularPeriodicContainer }n
  1{ O = PeriodName, t = "string<30>" }1 +
  1{ O = Period1stDate, t = "date" }1 +
  1{ O = Period1stTime, t = "time" }1 +
  1{ O = PeriodCycleDurationUnit, t = "string<15>" }1 +
  1{ O = PeriodCycleDurationValue, t = "real" }1 +
  1{ O = PeriodDurationUnit, t = "string<15>" }1 +
  1{ O = PeriodDurationValue, t = "real" }1 +
  1{ G = SingleDateTimes }1

1{ C = MultipleDateTimePeriodContainer }n
  1{ O = MultipleDateTimeName, t = "string<30>" }1 +
  1{ G = SingleDateTimes }1

0{ G = PlatformCharacteristic }1
  1{ C = PlatformCharacteristicContainer }n

0{ G = Instrument }1
  1{ C = InstrumentContainer }n

0{ G = PhysicalParameterDetails }1
  0{ O = ParameterUnitsOfMeasurement, t = "string<20>" }1 +
  0{ O = ParameterRangeBegin, t = "string<40>" }1 +
  0{ O = ParameterRangeEnd, t = "string<40>" }1 +
  0{ O = ParameterValueAccuracy, t = "string<30>" }1 +
  0{ O = ParameterValueAccuracyExplanation, t = "string<255>" }1 +
  0{ O = ParameterMeasurementResolution, t = "string<30>" }1

0{ G = InformationContent }1
  1{ O = ParameterValue, t = "string<255>" }1

```

### ***Collection Level Metadata Compound Definitions (Nest Level 5)***

```

0{ G = ContactPersonAddress }1
  0{ C = ContactPersonAddressContainer }n

0{ G = Telephone }1
  1{ C = TelephoneContainer }n

0{ G = Email }1
  1{ O = ElectronicEmailAddress, t = "string<255>" }1

0{ G = ContactOrganizationAddress }1
  1{ C = ContactOrganizationAddressContainer }n +

0{ G = OrganizationTelephone }1
  1{ C = OrganizationTelephoneContainer }n

0{ G = OrganizationEmail }1
  1{ O = ElectronicEmailAddress, t = "string<255>" }1

1{ C = VerticalSpatialDomainContainer }n
  1{ O = VerticalSpatialDomainType, t = "string<20>" }1 +
  1{ O = VerticalSpatialDomainValue, t = "string<20>" }1

0{ G = ZoneIdentifierClass }1
  1{ O = ZoneIdentifier, t = "string<64>" }1

1{ G = GPolygon }1
  1{ C = GPolygonContainer }n

1{ G = BoundingRectangle }1

```

```

1{ O = WestBoundingCoordinate, t = "real" }1 +
1{ O = NorthBoundingCoordinate, t = "real" }1 +
1{ O = EastBoundingCoordinate, t = "real" }1 +
1{ O = SouthBoundingCoordinate, t = "real" }1 +
1{ G = Point }1
 1{ O = PointLatitude, t = "real" }1 +
 1{ O = PointLongitude, t = "real" }1

1{ G = Circle }1
 1{ O = CenterLatitude, t = "real" }1 +
 1{ O = CenterLongitude, t = "real" }1 +
 1{ O = RadiusValue, t = "real" }1 +
 1{ O = RadiusUnits, t = "string<10>" }1

1{ G = AltitudeSystemDefinition }1
 1{ O = AltitudeDatumName, t = "string<40>" }1 +
 1{ O = AltitudeDistanceUnits, t = "string<20>" }1 +
 1{ O = AltitudeEncodingMethod, t = "string<255>" }1 +
 1{ O = AltitudeResolution, t = "real" }1

1{ G = DepthSystemDefinition }1
 1{ O = DepthDatumName, t = "string<80>" }1 +
 1{ O = DepthDistanceUnits, t = "string<20>" }1 +
 1{ O = DepthEncodingMethod, t = "string<255>" }1 +
 1{ O = DepthResolution, t = "real" }1

0{ G = GeodeticModel }1
 0{ O = HorizontalDatumName, t = "string<30>" }1 +
 1{ O = EllipsoidName, t = "string<255>" }1 +
 1{ O = SemiMajorAxis, t = "real" }1 +
 1{ O = DenominatorofFlatteningRatio, t = "real" }1

1{ G = GeographicCoordinateSystem }1
 1{ O = LatitudeResolution, t = "real" }1 +
 1{ O = LongitudeResolution, t = "real" }1 +
 1{ O = GeographicCoordinateUnits, t = "string<80>" }1

1{ G = PlanarCoordinateSystems }1
 1{ G = PlanarCoordinateSystem }1

1{ G = LocalCoordinateSystem }1
 1{ O = LocalCoordinateSystemDescription, t = "string<255>" }1 +
 1{ O = LocalGeoreferenceInformation, t = "string<255>" }1

1{ G = SingleDateTimes }1
 2{ C = SingleDateTimesContainer }n

1{ C = PlatformCharacteristicContainer }n
 1{ O = PlatformCharacteristicName, t = "string<40>" }1 +
 1{ O = PlatformCharacteristicDescription, t = "string<80>" }1 +
 1{ O = PlatformCharacteristicDataType, t = "string<8>" }1 +
 1{ O = PlatformCharacteristicUnit, t = "string<20>" }1 +
 1{ G = PlatformCharacteristicValueClass }1

1{ C = InstrumentContainer }n
 1{ O = InstrumentShortName, t = "string<20>" }1 +
 0{ O = InstrumentLongName, t = "string<80>" }1 +
 0{ O = InstrumentTechnique, t = "string<80>" }1 +

```

```

0{ O = NumberofSensors, t = "integer<>" }1 +
0{ G = OperationModeClass }1 +
0{ G = InstrumentCharacteristic }1 +
0{ G = Sensor }1

```

### ***Collection Level Metadata Compound Definitions (Nest Level 6)***

```

0{ C = ContactPersonAddressContainer }n
  1{ O = StreetAddress, t = "string<80>" }1 +
  1{ O = City, t = "string<30>" }1 +
  1{ O = StateProvince, t = "string<20>" }1 +
  1{ O = PostalCode, t = "string<20>" }1 +
  1{ O = Country, t = "string<20>" }1

1{ C = TelephoneContainer }n
  1{ O = TelephoneNumber, t = "string<23>" }1 +
  1{ O = TelephoneNumberType, t = "string<10>" }1

1{ C = ContactOrganizationAddressContainer }n
  1{ O = StreetAddress, t = "string<80>" }1 +
  1{ O = City, t = "string<30>" }1 +
  1{ O = StateProvince, t = "string<20>" }1 +
  1{ O = PostalCode, t = "string<20>" }1 +
  1{ O = Country, t = "string<10>" }1

1{ C = OrganizationTelephoneContainer }n
  1{ O = TelephoneNumber, t = "string<23>" }1 +
  1{ O = TelephoneNumberType, t = "string<10>" }1

1{ C = GPolygonContainer }n
  1{ G = GRing }1 +
  1{ G = GRingPoint }1

1{ G = PlanarCoordinateSystem }1
  1{ C = PlanarCoordinateSystemContainer }n

2{ C = SingleDateTimesContainer }n
  1{ O = TimeOfDay, t = "time" }1 +
  1{ O = CalendarDate, t = "date" }1

1{ G = PlatformCharacteristicValueClass }1
  1{ O = PlatformCharacteristicValue, t = "string<255>" }1

0{ G = OperationModeClass }1
  1{ O = OperationMode, t = "string<20>" }n

0{ G = InstrumentCharacteristic }1
  1{ C = InstrumentCharacteristicContainer }n

0{ G = Sensor }1
  1{ C = SensorContainer }n

```

### ***Collection Level Metadata Compound Definitions (Nest Level 7)***

```

1{ G = GRing }1
  1{ O = ExclusionGRingFlag, t = "string<1>" }1

1{ G = GRingPoint }1
  3{ O = GRingPointLatitude, t = "real" }n +
  3{ O = GRingPointLongitude, t = "real" }n +

```

```

3{ O = GRingPointSequenceNo, t = "integer<>" }n
1{ C = PlanarCoordinateSystemContainer }n
  1{ G = PlanarCoordinateInformation }1 +
  1[ 1{ G = MapProjection }1 | 1{ G = LocalPlanarCoordinateSystem }1 | 1{ G = GridCoordinateSystem }1 ]1
1{ C = InstrumentCharacteristicContainer }n
  1{ O = InstrumentCharacteristicName, t = "string<40>" }1 +
  1{ O = InstrumentCharacteristicDescription, t = "string<80>" }1 +
  1{ O = InstrumentCharacteristicUnit, t = "string<20>" }1 +
  1{ O = InstrumentCharacteristicDataType, t = "string<8>" }1 +
  1{ G = InstrumentCharacteristicValueClass }1
1{ C = SensorContainer }n
  1{ O = SensorShortName, t = "string<20>" }1 +
  0{ O = SensorLongName, t = "string<80>" }1 +
  1{ O = SensorTechnique, t = "string<80>" }1 +
  0{ G = SensorCharacteristic }1

```

### ***Collection Level Metadata Compound Definitions (Nest Level 8)***

```

1{ G = PlanarCoordinateInformation }1
  1{ O = PlanarDistanceUnits, t = "string<80>" }1 +
  1{ O = PlanarCoordinateEncodingMethod, t = "string<80>" }1 +
  1[ 1{ G = DistanceandBearingRepresentation }1 | 1{ G = CoordinateRepresentation }1 ]1
1{ G = MapProjection }1
  1{ O = MapProjectionName, t = "string<80>" }1 +
  0{ O = MapProjectionPointer, t = "string<255>" }1 +
1{ G = LocalPlanarCoordinateSystem }1
  1{ O = LocalPlanarCoordinateSystemDescription, t = "string<255>" }1 +
  1{ O = LocalPlanarGeoreferenceInformation, t = "string<255>" }1 +
1{ G = GridCoordinateSystem }1
  1{ O = GridCoordinateSystemName, t = "string<255>" }1 +
1{ G = InstrumentCharacteristicValueClass }1
  1{ O = InstrumentCharacteristicValue, t = "string<255>" }1 +
0{ G = SensorCharacteristic }1
  1{ C = SensorCharacteristicContainer }n

```

### ***Collection Level Metadata Compound Definitions (Nest Level 9)***

```

1{ G = DistanceandBearingRepresentation }1
  1{ O = DistanceResolution, t = "real" }1 +
  1{ O = BearingResolution, t = "real" }1 +
  1{ O = BearingUnits, t = "string<255>" }1 +
  1{ O = BearingReferenceDirection, t = "string<20>" }1 +
  1{ O = BearingReferenceMeridian, t = "string<255>" }1
1{ G = CoordinateRepresentation }1
  1{ O = AbscissaResolution, t = "real" }1 +
  1{ O = OrdinateResolution, t = "real" }1
1{ C = SensorCharacteristicContainer }n
  1{ O = SensorCharacteristicName, t = "string<10>" }1 +
  1{ O = SensorCharacteristicDescription, t = "string<80>" }1 +

```

```
1{ O = SensorCharacteristicDataType, t = "string<8>" }1 +
0{ O = SensorCharacteristicUnit, t = "string<20>" }1 +
1{ G = SensorCharacteristicValueClass }1
```

### ***Collection Level Metadata Compound Definitions (Nest Level 10)***

```
1{ G = SensorCharacteristicValueClass }1
    1{ O = SensorCharacteristicValue, t = "string<255>" }1
```

## **Granule Level Inventory Metadata**

For the "full metadata coverage" all optional classes and attributes that are relevant to the collection will be provided. For example, though the "InputGranule" group is listed as 0{ }1, indicating that the group is optional, all EOS Standard Products above level 0 will provide this group.

### ***Full Granule Level Inventory Metadata Coverage***

```
1{ M = InventoryMetadata }1
    1{ G = CollectionDescriptionClass }1 +
    1{ G = ECSDataGranule }1 +
    1[ 1{ G = SingleDateTime }1 / 1{ G = RangeDateTime }1 ]1 +
    0{ G = MeasuredParameter }1 +
    0{ G = OrbitCalculatedSpatialDomain }1 +
    0{ G = InputGranule }1 +
    0{ G = SpatialDomainContainer }1 +
    0{ G = PGEVersionClass }1 +
    0{ G = AncillaryInputGranule }1 +
    0{ G = Review }1 +
    0{ G = ProcessingQA }1 +
    0{ G = AdditionalAttributes }1 +
    0{ G = OrbitParametersGranule }1 +
    0{ G = StorageMediumClass }1 +
    0{ G = AnalysisSource }1 +
    0{ G = Campaign }1 +
    0{ G = SensorCharacteristic }1 +
    0{ G = AssociatedPlatformInstrumentSensor }1
```

### ***Intermediate Granule Level Inventory Metadata Coverage***

```
1{ M = InventoryMetadata }1
    1{ G = CollectionDescriptionClass }1 +
    1{ G = ECSDataGranule }1 +
    0[ 1{ G = SingleDateTime }1 / 1{ G = RangeDateTime }1 ]1 +
    0{ G = MeasuredParameter }1 +
    0{ G = OrbitCalculatedSpatialDomain }1 +
    0{ G = InputGranule }1 +
    0{ G = SpatialDomainContainer }1 +
    0{ G = PGEVersionClass }1 +
    0{ G = AncillaryInputGranule }1 +
    0{ G = Review }1 +
    0{ G = ProcessingQA }1 +
    0{ G = AdditionalAttributes }1 +
    0{ G = OrbitParametersGranule }1 +
    0{ G = StorageMediumClass }1 +
```

```

0{ G = AnalysisSource }1 +
0{ G = Campaign }1 +
0{ G = SensorCharacteristic }1 +
0{ G = AssociatedPlatformInstrumentSensor }1

```

### **Limited or Minimal Granule Level Inventory Metadata Coverage**

```

1{ M = InventoryMetadata }1
  1{ G = CollectionDescriptionClass }1 +
  0{ G = ECSDataGranule }1 +
  0[ 1{ G = SingleDateTime }1 / 1{ G = RangeDateTime }1 ]1 +
  0{ G = MeasuredParameter }1 +
  0{ G = OrbitCalculatedSpatialDomain }1 +
  0{ G = InputGranule }1 +
  0{ G = SpatialDomainContainer }1 +
  0{ G = PGEVersionClass }1 +
  0{ G = AncillaryInputGranule }1 +
  0{ G = Review }1 +
  0{ G = ProcessingQA }1 +
  0{ G = AdditionalAttributes }1 +
  0{ G = OrbitParametersGranule }1 +
  0{ G = StorageMediumClass }1 +
  0{ G = AnalysisSource }1 +
  0{ G = Campaign }1 +
  0{ G = SensorCharacteristic }1 +
  0{ G = AssociatedPlatformInstrumentSensor }1 +
  0[ 1{ V = DAP }1 | 1{ V + AP }1 / 1{ V = SSAP }1 ]1

```

### **Granule Level Inventory Metadata Compound Definitions (Nest level 2)**

```

1{ G = CollectionDescriptionClass }1
  1{ O = ShortName, t = "string<8>" }1 +
  1{ O = VersionID, t = "integer<0:255>" }1

1{ G = ECSDataGranule }1
  1{ O = SizeMBECSDataGranule, t = "real" }1 +
  0{ O = ReprocessingPlanned, t = "string<20>" }1 +
  0{ O = ReprocessingActual, t = "string<255>" }1 +
  0{ O = LocalGranuleID, t = "string<80>" }1 +
  0{ O = DayNightFlag, t = "string<5>" }1 +
  1{ O = ProductionDateTime, t = "date_time" }1 +
  0{ O = LocalVersionID, t = "string<60>" }1

1{ G = SingleDateTime }1
  1{ O = TimeOfDay, t = "time" }1 +
  1{ O = CalendarDate, t = "date" }1

1{ G = RangeDateTime }1
  1{ O = RangeBeginningTime, t = "time" }1 +
  1{ O = RangeEndingTime, t = "time" }1 +
  1{ O = RangeBeginningDate, t = "date" }1 +
  1{ O = RangeEndingDate, t = "date" }1

0{ G = MeasuredParameter }1
  1{ C = MeasuredParameterContainer }m

```

```

0{ G = OrbitCalculatedSpatialDomain }1
  1{ C = OrbitCalculatedSpatialDomainContainer }m

0{ G = InputGranule }1
  1{ O = InputPointer, t = "string<255>" }n

0{ G = SpatialDomainContainer }1
  0{ G = GranuleLocality }1 +
  0{ G = VerticalSpatialDomain }1 +
  0{ G = HorizontalSpatialDomainContainer }1 +

0{ G = PGEVersionClass }1
  1{ O = PGEVersion, t = "string<10>" }1

0{ G = AncillaryInputGranule }1
  1{ C = AncillaryInputGranuleContainer }m

0{ G = Review }1
  1{ C = ReviewContainer }m

0{ G = ProcessingQA }1
  0{ C = ProcessingQAContainer }m

0{ G = AdditionalAttributes }1
  1{ C = AdditionalAttributesContainer }m

0{ G = OrbitParametersGranule }1
  1{ O = OrbitParametersPointer, t = "string<255>" }n

0{ G = StorageMediumClass }1
  1{ O = StorageMedium, t = "string<30>" }n

0{ G = AnalysisSource }1
  1{ O = AnalysisShortName, t = "string<20>" }n

0{ G = Campaign }1
  1{ O = CampaignShortName, t = "string<20>" }n

0{ G = SensorCharacteristic }1
  0{ C = SensorCharacteristicContainer }m

0{ G = AssociatedPlatformInstrumentSensor }1
  1{ C = AssociatedPlatformInstrumentSensorContainer }m

1{ V = DAP }1
  1{ O = DAPID, t = "string<n>" }1 +
  1{ O = DAPIsertDate, t = "date" }1 +
  1{ G = PGEGroups }1

1{ V = AP }1
  1{ O = AlgorithmPackageName, t = "string<80>" }1 +
  1{ O = AlgorithmPackageVersion, t = "string<n>" }1 +
  1{ O = AlgorithmPackageMaturityCode, t = "string<10>" }1 +
  1{ O = AlgorithmPackageAcceptanceDate, t = "date_time" }1 +
  1{ O = DeliveryPurpose, t = "string<20>" }1 +
  1{ O = PGENAME, t = "string<20>" }1 +
  1{ O = PGEVersion, t = "string<10>" }1 +
  1{ O = PGEIdentifier, t = "string<10>" }1 +
  1{ O = PGFunction, t = "string<80>" }1 +
  1{ O = PGEDateLastModified, t = "date_time" }1 +
  1{ O = SWVersion, t = "string<12>" }1 +

```

```

1{ O = SWDateLastModified, t = "date_time" }1 +
1{ G = AssociatedCollections }1

1{ V = SSAP }1
  1{ O = ComponentType, t = "string<40>" }1 +
  1{ O = ComponentName, t = "string<80>" }1 +
  1{ O = SSAPAlgorithmPackageName, t = "string<80>" }1 +
  1{ O = SSAPInsertDate, t = "date_time" }1 +
  1{ G = AlgorithmPackageVersions }1

```

### ***Granule Level Inventory Metadata Compound Definitions (Nest level 3)***

```

1{ C = MeasuredParameterContainer }m
  1{ O = ParameterName, t = "string<40>" }1 +
  1{ G = QAFlags }1 +
  0{ G = QAStats }1

1{ C = OrbitCalculatedSpatialDomainContainer }m
  0{ O = OrbitalModelName, t = "string<80>" }1 +
  0[1{ O = OrbitNumber, t = "integer<i:j>" }1 / (1{ O = StartOrbitNumber, t = "integer<i:j>" }1 + 1{ O =
  StopOrbitNumber, t = "integer<i:j>" }1) ]1 +
  1{ O = EquatorCrossingLongitude, t = "real" }1 +
  1{ O = EquatorCrossingTime, t = "time" }1 +
  1{ O = EquatorCrossingDate, t = "date" }1

0{ G = GranuleLocality }1
  0{ O = LocalityValue, t = "string<80>" }n

0{ G = VerticalSpatialDomain }1
  0{ C = VerticalSpatialDomainContainer }m

0{ G = HorizontalSpatialDomainContainer }1
  0{ G = ZoneIdentifierClass }1 +
  1[ G = 1{ GPolygon }1 / 1{ G = BoundingRectangle }1 / 1{ G = Point }1 / 1{ G = Circle }1 ]1

1{ C = AncillaryInputGranuleContainer }m
  1{ O = AncillaryInputType, t = "string<20>" }n +
  1{ O = AncillaryInputPointer, t = "string<255>" }n

1{ C = ReviewContainer }m
  1{ O = ScienceReviewDate, t = "date" }1
  1{ O = ScienceReviewStatus, t = "string<255>" }1
  0{ O = FutureReviewDate, t = "date" }1

0{ C = ProcessingQAContainer }m
  1{ O = ProcessingQADescription, t = "string<255>" }1 +
  1{ O = ProcessingQAAttribute, t = "string<80>" }1

1{ C = AdditionalAttributesContainer }m
  1{ O = AdditionalAttributesName, t = "string<30>" }1 +
  1{ G = InformationContent }1

0{ C = SensorCharacteristicContainer }m
  1{ O = PlatformShortName, t = "string<20>" }1 +
  1{ O = InstrumentShortName, t = "string<20>" }1 +
  1{ O = SensorShortName, t = "string<80>" }1 +
  1{ O = SensorCharacteristicName, t = "string<80>" }1 +
  1{ O = SensorCharacteristicValue, t = "string<20>" }1

```

```

1{ C = AssociatedPlatformInstrumentSensorContainer }m
  1{ O = AssociatedPlatformShortName, t = "string<n>" }1 +
  1{ O = AssociatedInstrumentShortName, t = "string<n>" }1 +
  1{ O = AssociatedSensorShortName, t = "string<n>" }1 +
  0{ O = OperationMode, t = "string<20>" }1

1{ G = PGEGroups }1
  1{ C = PGEGroupContainer }m

1{ G = AssociatedCollections }1
  1{ C = AssociatedCollectionContainer }m

1{ G = AlgorithmPackageVersions }1
  1{ C = AlgorithmPackageVersionContainer }m

```

### ***Granule Level Inventory Metadata Compound Definitions (Nest level 4)***

```

1{ G = QAFlags }1
  1{ O = AutomaticQualityFlag, t = "string<64>" }1 +
  1{ O = AutomaticQualityFlagExplanation, t = "string<255>" }1 +
  0{ O = OperationalQualityFlag, t = "string<20>" }1 +
  0{ O = OperationalQualityFlagExplanation, t = "string<255>" }1 +
  0{ O = ScienceQualityFlag, t = "string<20>" }1 +
  0{ O = ScienceQualityFlagExplanation, t = "string<255>" }1

0{ G = QAStats }1
  1{ O = QAPercentMissingData, t = "integer<i:j>" }1 +
  0{ O = QAPercentInterpolatedData, t = "integer<i:j>" }1 +
  0{ O = QAPercentOutofBoundsData, t = "integer<i:j>" }1 +
  0{ O = QAPercentCloudCover, t = "integer<i:j>" }1

0{ C = VerticalSpatialDomainContainer }m
  0{ O = VerticalSpatialDomainType, t = "string<20>" }1
  0{ O = VerticalSpatialDomainValue, t = "string<20>" }1

0{ G = ZoneIdentifierClass }1
  1{ O = ZoneIdentifier, t = "string<64>" }1

1{ G = GPolygon }1
  1{ C = GPolygonContainer }m

1{ G = BoundingRectangle }1
  1{ O = WestBoundingCoordinate, t = "real" }1 +
  1{ O = NorthBoundingCoordinate, t = "real" }1 +
  1{ O = EastBoundingCoordinate, t = "real" }1 +
  1{ O = SouthBoundingCoordinate, t = "real" }1

1{ G = Point }1
  1{ O = PointLongitude, t = "real" }1
  1{ O = PointLatitude, t = "real" }1

1{ G = Circle }1
  1{ O = CenterLatitude, t = "real" }1 +
  1{ O = CenterLongitude, t = "real" }1 +
  1{ O = RadiusValue, t = "real" }1 +
  1{ O = RadiusUnits, t = "string<10>" }1

1{ O = InformationContent }1
  1{ G = ParameterValue, t = "string<10>" }n

```

```

1{ C = PGEGroupContainer }m
  1{ O = DAPPGEName, t = "string<n>" }1 +
  1{ O = DAPPGEVersion, t = "string<n>" }1 +
  1{ O = DAPSWVersion, t = "string<n>" }1

1{ C = AssociatedCollectionContainer }m
  1{ O = APCollectionShortName, t = "string<n>" }1 +
  1{ O = APCollectionVersionID, t = "string<n>" }1

1{ C = AlgorithmPackageVersionContainer }m
  1{ O = SSAPAlgPackageVersion, t = "string<20>" }1 +

```

### ***Granule Level Inventory Metadata Compound Definitions (Nest level 5)***

```

1{ C = GPolygonContainer }m
  1{ G = GRing }1 +
  1{ G = GRingPoint }1

```

### ***Granule Level Inventory Metadata Compound Definitions (Nest level 6)***

```

1{ G = GRing }1
  1{ O = ExclusionGRingFlag, t = "string<1>" }1

1{ G = GRingPoint }1
  3{ O = GRingPointLatitude, t = "real" }n +
  3{ O = GRingPointLongitude, t = "real" }n +
  3{ O = GRingPointSequenceNo, t = "integer<i:j>" }n

```

## **Syntax used in this definition:**

<X> = element, where <X> is one of:

### **M Master Group:**

A Master Group is an ODL group that contains a "GROUPTYPE = MASTER" statement

### **G Group**

A Group is an ODL structure that contains one or more nested structures and no other ODL statements. Groups never repeat within the same nest level.

### **V Virtual Group**

These are "virtual groups" that apply to specific ESDTs (only DAP, AP, and SSAP). The corresponding GROUP and ENDGROUP statements in the ODL must be enclosed by comment markers /\* and \*/

### **C Container Object**

Container Objects are ODL Objects that contain one or more nested structures. The Container Object is a mechanism for repeating complex structures at a particular nest level because the Group structure is never repeated. Container Objects contain the following statements in addition to the structures that they contain:

Data\_Location = "l"

Mandatory = "m"

Where, "l" is the location of the "Value" statement and "m" is either "TRUE" or "FALSE" and indicates whether the value must be obtained. The meaning of these statements is discussed elsewhere [TBD]. Note to ECS: It is not obvious why Container Objects need these statements since they do not correspond to data model attributes and therefore have no "value".

Container Objects and the immediately subordinate structures that they contain must also include within them the statement

Class = "M"

Where, "M" is the number of times the Container Object is repeated.

### **O Simple Object**

A Simple Object is an ODL Object that does not contain and subordinate structures. It is the "leaf" level ODL structure in the data model implementation. Each Simple Object corresponds to one and only one attribute in the data model. All Simple Objects in the MCF contain the following ODL statements:

Data\_Location = "l"

Mandatory = "m"

Type = "t"

Num\_Val = "n"

See elsewhere for a discussion of the meaning of "l" and "m". "t" is the type of the data and "n" is the maximum number of values that may be provided in the corresponding metadata ODL "Value" statement. "n" may never be zero; if zero values are desired, the Object must be omitted. If "n" is greater than one, the Object is multivalued and the corresponding metadata ODL "Value" statement must contain an array.

Additionally, if the Simple Object is nested immediately subordinate to a Container Object, the associated MCF must contain an ODL statement:

Class = "M"

Where, "M" is the number of times the Container Object is repeated.

**t Type**

Found in the definition next to Simple Objects, it is the valid type of the corresponding metadata ODL "VALUE" statement. ODL is text, so all "types" are sequences of characters. Type is one of:

string<n> Text enclosed in double quotes, no longer than "n" characters (not including the quote characters. For example: "string<25>":

*VALUE = "EOS Sensor"*

real Numeric representation of a real number, either represented in a decimal notation or in a scientific notation with a base 10 exponent specified. May include a number sign, unsigned real numbers are positive. For example "real":

*VALUE = 123.4*

*VALUE = -1.234E+3*

integer<i:j> Numeric representation of an integer in the range from i to j inclusive; a string of digits, optionally preceded by a number sign. For example: "integer<0:255>"

*VALUE = 123*

date calendar date in the form yyyy:mm:dd, where

yyyy the four digit year AD

mm the two digit month in the range 01 <= mm >= 12

dd the integral day of month in the range appropriate for the given month

The ":" and "." punctuation marks are literal.

or alternately in the form yyyy:doy

yyyy the four digit year AD

doy the integral day of year in the range appropriate for the given year

The ":" punctuation mark is literal.

For example:

*VALUE = 1999:02:03*

*VALUE = 1999:34*

time UTC time in the form hh:mm:ss.ssssZ, where:

hh integral number of hours in the range 00 = < hh < 24

mm integral number of minutes in the range from 00 = < mm < 60

ss.ssss seconds in the range 00 = < ss.ssss > 60(either an integral or a fractional number of seconds can be specified)

z time zone: if Greenwich use z is "Z", use 1{ +|- }1+zz, where zz is the number of hours offset from Greenwich

The ":" and "." punctuation marks are literal

For example:

*VALUE = 01:10:39.35Z*

*VALUE = 23:59:59-07*

**date\_time** Combined date and time using the formats . The date first, followed by the letter "T" followed by the time using the definitions of date and time given above. For example:

*VALUE = 1999:02:03T14:11:35-05*

*VALUE = 1999:02:03T19:11:35Z*

**Element is one of:**

**i{ element }j**

**the element is repeated from i to j times**

If  $i = 0$ , the element (and its nested components) are optional.

If  $i = 1$ , the element is required (if its higher level parent occurs).

If  $j = 1$ , there may be only one occurrence of this element.

If  $j = m$ , the element is a "Class = M" container object (see **C** above).

If  $j = n$ , the element is a multivalued array object.

**i[ element1 | element2 ]j** Choose one from element1 or element2 from  $i$  to  $j$  times

**Symbols:**

**+**

**Read as "and" (on the same nest level)**

**=**

**Read as "is"**